

# **Absolvování individuální odborné praxe**

## **Individual Professional Practice in the Company**

## Zadání bakalářské práce

Student: **Michal Balický**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**  
**Individual Professional Practice in the Company**

### Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: GARVIS Solutions s.r.o.
2. Struktura závěrečné zprávy:
  - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
  - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
  - c) Zvolený postup řešení zadaných úkolů.
  - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
  - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
  - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

### Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Petr Gajdoš, Ph.D.**

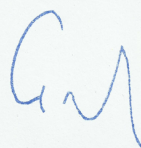
Konzultant bakalářské práce: Ing. Radka Radecká

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty



## Prohlášení zástupce spolupracující právnické osoby

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.



**GARVIS Solutions s.r.o.**

Studentská 6202/17  
708 00 Ostrava - Poruba


IC: 23059638 DIČ: CZ28659058  
OR u KS v Ostravě, oddíl C, vložka 37572

.....  
Ing. Michal Radecký, Ph.D.

V Ostravě dne 30.4. 2015

## Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.



.....  
Michal Balický

V Ostravě dne 27. dubna 2015



Rád bych poděkoval kolektivu společnosti GARVIS Solutions, s.r.o. a vedoucímu mé bakalářské práce, Ing. Petru Gajdošovi, Ph.D.

## Abstrakt

Tato bakalářská práce se zabývá převážně popisem a průběhem praxe ve společnosti GARVIS Solutions, s.r.o. Počátek samotné práce je otevírán popisem zmíněné společnosti. Dále následuje výčet použitých technologií, jejich aspekty a zdůvodnění jejich výběru. Následuje pojednání o samotné práci na aplikaci Čistota ovzduší a jednotlivých jejích součástech. V této části práce je zmíněno celé spektrum úkonů, od přepracovávání kódu z původně použitých platforem, a jeho migrace, po využití nejnovějších technologií Windows Runtime, se zapojením map Bing. V závěrečné části pojednávám o znalostech a zkušenostech, které jsem v průběhu práce využil, a naopak, které mi chyběly a musel jsem je nabýt.

**Klíčová slova:** Windows, Univerzální aplikace, Windows Runtime, C#, GARVIS Solutions, Čistota ovzduší

## Abstract

This bachelor work deals mainly with description and proces of professional practice in GARVIS Solutions, s.r.o company. The beginning of the work itself is opened by description of mentioned company. This is followed by a list of technologies, their aspects and rationale for their selection. Follows a thorough discussion of the actual work on the application of Pollution CZ and its individual components. In this part of the work is mentioned the spectrum of operations, from reworking code from originally used platforms, and its migration, to the utilization of the latest technologies of Windows Runtime, involving Bing maps. In the final part I deal with the knowledge and experience that I have used in the course of work, and vice versa, which I missed and I had to take.

**Keywords:** Windows, Universal Apps, Windows Runtime, C#, GARVIS Solutions, Air pollution



## Seznam použitých zkratk a symbolů

API	– Application Programming Interface
CMS	– Content Management System
ČHMÚ	– Český Hydrometeorologický Ústav
GPS	– Global Positioning System
MVVM	– Model View ViewModel
WinRT	– Windows Runtime
OS	– Operační systém
UI	– User Interface
WPF	– Windows Presentation Foundation
XAML	– Extensible Application Markup Language

## Obsah

<b>1</b>	<b>Úvod</b>	<b>4</b>
1.1	GARVIS Solutions, s.r.o. . . . .	4
<b>2</b>	<b>Obsah praxe</b>	<b>6</b>
2.1	WinRT . . . . .	6
2.2	Windows Universal Apps . . . . .	6
2.3	Práce na aplikaci Čistota ovzduší . . . . .	7
<b>3</b>	<b>Vývoj aplikace a práce s WinRT</b>	<b>8</b>
3.1	Migrace kódu z verze 8 na verzi 8.1 . . . . .	8
3.2	MVVM . . . . .	8
3.3	Uživatelské rozhraní . . . . .	9
3.4	Využití navigace . . . . .	12
3.5	Úlohy na pozadí a živé dlaždice . . . . .	13
3.6	Komunikace . . . . .	15
3.7	Využití vlastností architektury MVVM a jazyka XAML . . . . .	15
3.8	Práce s vlákny a asynchronními operacemi . . . . .	16
3.9	Dokončení aplikace . . . . .	17
<b>4</b>	<b>Poznámky k platformě a zhodnocení znalostí</b>	<b>19</b>
4.1	Uplatněné znalosti . . . . .	19
4.2	Scházející a nabyté znalosti . . . . .	20
<b>5</b>	<b>Závěr</b>	<b>22</b>
<b>6</b>	<b>Reference</b>	<b>23</b>



## Seznam obrázků

1	Úvodní stránka aplikace . . . . .	9
2	Informační panel . . . . .	10
3	Panel mapy - slepá mapa . . . . .	10
4	Panel mapy - detailní mapa . . . . .	11
5	Stránka stanice . . . . .	11
6	Statistika stahovanosti . . . . .	18

## Seznam výpisů zdrojového kódu

1	Element pro zobrazení hodnoty jednoho z prvků . . . . .	12
2	Část metody pro obsloužení události změny geografické polohy . . . . .	13
3	Naplnění a aktualizace dlaždice . . . . .	14
4	Přiřazení části kódu UI vláknu . . . . .	16
5	Korektní hlavička asynchronní metody . . . . .	16



## 1 Úvod

Praxi ve firmě jsem si vybral, abych mohl své teoretické znalosti, nabyté studiem ve škole i v soukromí, doplnit o praktické zkušenosti a obě porovnat. Také jsem si chtěl doplnit znalosti a zkušenosti i o oblasti, které mi škola už ze své povahy nemohla nabídnout.

Vždy jsem se zajímal o vývoj aplikací a dlouho jsem tedy hledal firmu, kterou bych mohl oslovit ohledně absolvování bakalářské praxe. Nakonec jsem našel GARVIS Solutions, kteří se, mimo jiné, vývojem aplikací zabývají. Shodou okolností jsem se ve volném čase učil na tutoriálech pro vývoj aplikací na Windows Phone, které jsou vytvořeny a publikovány právě touto společností.

Před oslovením GARVIS Solutions jsem sice měl jisté zkušenosti s aplikacemi, ale touto cestou jsem se mohl dostat k vývoji opravdu rozsáhlejšího projektu, a navíc pod odborným dohledem. S nadšením jsem tedy nastoupil na praxi, která mimo klasické práce v kanceláři, zahrnovala i práci z domu. Nespornou výhodou je umístění kanceláře společnosti v kampusu univerzity, kam jsem docházel kvůli práci a kvůli konzultacím.

V druhé kapitole se zabývám praxí samotnou a jejím průběhem.

V třetí kapitole pojednávám o platformě, se kterou jsem pracoval. O jejích silných a slabých stránkách.

Čtvrtá kapitola je věnována zpětné vazbě, tedy zhodnocení celé praxe, mé připravenosti k jejímu absolvování a zkušenostem, které jsem díky ní získal.

### 1.1 GARVIS Solutions, s.r.o.

Firma GARVIS Solutions s.r.o. byla založena roku 2010 a sídlí v areálu Vysoké školy báňské – Technické univerzity Ostrava na adrese Studentská 6202/17. Ze začátku byla orientována především na webové technologie. Dnes nabízí také další služby v oblasti informačních technologií, grafiky či marketingu. Jedná se o malou společnost, která si zakládá na zkušenostech a profesionalitě, ale také na inovativním a osobním přístupu ke každému projektu [1].

Hlavním projektem společnosti je redakční systém Nestigo, který si dává za cíl poskytnout klientovi možnosti velkých, profesionálně zpracovaných webových aplikací, přičemž zachovává důraz na specifické požadavky a velkou míru personalizace. Jedná se o CMS poskytované zákazníkům formou služby. Obsahuje celou řadu funkčních modulů, podporu uživatelských rolí, multijazyčnost, propojení se sociálními sítěmi a další funkce. Jedná se tedy o vysoce univerzální nástroj pro vytvoření a jednoduchou správu webové prezentace.

Mimo systému Nestigo má za sebou GARVIS Solutions bohaté portfolio, v němž nechybí ani již zmiňované mobilní aplikace. Lze uvést například aplikace Čistota ovzduší, Stopky, Svátky, nebo Hrací kostka, realizovaná v Direct3D. Všechny tyto aplikace jsou k dispozici na Windows Store pro Windows Phone.

## 2 Obsah praxe

Jak bylo řečeno v úvodu, společnost GARVIS Solutions se zabývá také mobilními aplikacemi. Jedním z těchto projektů je mobilní aplikace Čistota ovzduší. Ta je k dispozici pro platformu Windows Phone verzí 7.5, 8, 8.1. Slouží k rychlému zjištění kvality ovzduší a jeho znečištění na všech místech České Republiky, kde se zároveň nachází měřicí stanice ČHMÚ. Také umožňuje lokalizaci telefonu spolu s přidruženými službami a sdílení stavu ovzduší nejbližší stanice, který sestává ze sdílené fotografie, slovního hodnocení a číselného hodnocení.

Hlavní náplní mé odborné praxe bylo vytvořit aplikaci Čistota ovzduší pro Windows 8 a Windows 8.1. K dispozici jsem dostal zdrojový kód mobilní aplikace. Ten nicméně musel být takřka kompletně přepracován kvůli využití rozdílných technologií a platform, než byly využity u předchozí mobilní verze. Také byl zvolen diametrálně odlišný přístup k uživatelskému rozhraní aplikace.

Původní kód byl řešen v softwarové platformě .NET. Má práce spočívala ve vypracování projektu v architektuře Windows Runtime, dále jen WinRT. Znamenalo to téměř kompletně přepracovat stávající aplikaci a využít jiných, či nových technologií, které tato architektura nabízí. Z části také proto, že WinRT nepodporuje většinu platform a frameworků do té doby používaných k vývoji aplikací pro operační systémy od společnosti Microsoft [2].

### 2.1 WinRT

Windows Runtime, nebo-li zkráceně WinRT, bylo představeno poprvé v operačním systému Windows Server 2012 a mimo jiné je zahrnut ve Windows 8/8.1 a Windows 10. Co se týče Windows Phone 8/8.1, zde je k dispozici Windows Phone Runtime. Obě architektury jsou si ovšem záměrně velice podobné, což umožňuje nedělat při vývoji programů pro tyto dva operační systémy velké rozdíly. WinRT obecně podporuje vývoj v několika jazycích, například C#, C++, VB .NET. Poskytuje mnohá API pro vývoj Windows Store, Windows Phone aplikací a Windows Universal Apps.

V desktopových a mobilních aplikacích je doplněno o Modern UI, jenž je známější pod pojmem Metro. Tento typ uživatelského rozhraní klade důraz na jednoduchost, striktní geometrické tvary a typografii.

### 2.2 Windows Universal Apps

Pojem univerzálních aplikací se dostal do vývoje programů pro Windows v roce 2013 s novou verzí operačního systému Windows 8.1 a Windows Phone 8.1. Microsoft se dlouhodobě snaží o těsné propojení počítačů, tabletů, chytrých telefonů a dalších zařízení,

kteřa jsou obsluhována operačním systémem z rodiny Windows NT. Díky tomu mohl vzniknout nový typ aplikací, takzvané univerzální aplikace. Tyto aplikace jsou označovány obchodním názvem Windows Universal Apps. Do té doby bylo možné se s podobným principem setkat pouze u produktů společnosti Apple.

Univerzální aplikace není z pohledu uživatele nijak zvláštní nebo převratná. Ve Windows Store si pořídí aplikaci, která mu může být k dispozici na všechna jeho zařízení. Tato zařízení mohou sdílet nastavení a případně i data.

Z hlediska vývoje ale univerzální aplikace přináší mnohem více. Vývojová šablona má sdílenou část, která je určena pro kód společný všem platformám. Další části jsou určeny pro komponenty jednotlivých cílových platform. V současnosti jsou to Windows 8.1 a Windows Phone 8.1. V těchto oddílech se nachází zpravidla UI a kód typický pro dané platformy, jelikož ne všechna API určená pro Windows jsou dostupná také pro Windows Phone a obráceně. Vývojář tedy tvoří jeden kód, který je částečně sdílený a vytváří z něj dvě sestavení.

S nadcházejícím Windows 10 by mělo přijít ale v tomto směru mnoho změn. Jelikož u této verze operačního systému nebude rozlišována platforma, budou moci být univerzální aplikace opravdu univerzální v pravém slova smyslu. Vývojář bude tvořit jeden kód, který je, včetně UI, společný pro všechny platformy a z něj vytvoří pouze jedno sestavení použitelné pro jakékoliv zařízení využívající Windows 10.

## 2.3 Práce na aplikaci Čistota ovzduší

Mým hlavním úkolem během odborné praxe bylo vytvořit aplikaci Čistota ovzduší, anglicky Pollution CZ (registrované názvy). Tato aplikace byla pod hlavičkou GARVIS Solutions k dostání na Windows Store již dříve a to ve verzi Windows Phone. Mým úkolem bylo zpracovat aplikaci pro operační systém Windows, tedy jako Windows Store App. Aplikace měla mít nové, vzdušnější rozhraní, při zachování všech důležitých dosavadních funkcí. Využil jsem možnosti vytvářet aplikaci jako Windows Universal App, ač bylo v záměru pracovat jen na desktopové verzi. Tento způsob umožní v budoucnu výrazné zjednodušení případné migrace nebo nového zpracování mobilní verze aplikace. Kód pro zpracování dat a většina algoritmů může být sdílen, o což je případný vývoj usnadněn. Do jisté míry je také možné sdílet i uživatelské rozhraní, které využívá prvky společné pro desktopovou i mobilní architekturu.

Aplikaci jsem vyvíjel v prostředí Visual Studio 2013 Ultimate a Blend for Visual Studio 2013.

## 3 Vývoj aplikace a práce s WinRT

Původní aplikace byla psána v .NET. Windows Runtime využívá ovšem vlastních API. Nebylo tedy možné převzít kód aplikace a vytvořit pouze uživatelské rozhraní pro Windows. Kód musel být migrován, a z velké části rovnou nahrazen novým. Ač je ze strany Microsoftu viditelná snaha o usnadnění migrace na WinRT z dřívějších frameworků, v mnoha případech je nutno zvolit diametrálně odlišný přístup.

### 3.1 Migrace kódu z verze 8 na verzi 8.1

Změna podporovaných technologií pro Windows a Windows Phone byla velkým krokem pro celou komunitu vývojářů Windows aplikací. Ovšem proměna, kterou prošel vývoj při přechodu z Windows Phone 8 na Windows Phone 8.1 byla podobně rozsáhlá. Pro Windows Phone 8.1 jsou změněna prakticky všechna API. Kupříkladu, k podstatné změně došlo ve využití GPS navigace, úloh na pozadí, vláken, asynchronního volání nebo práce se sítí. Kdybych měl popsat změnu svými slovy, řekl bych, že vývoj kódu na Windows Phone 8 může probíhat v čistokrevném .NET a je velice podobný vývoji WPF nebo do jisté míry i Silverlight aplikace, kdežto vývoj pro verzi 8.1 má vlastní pravidla. Využívá nejnovějších verzí C# a .NET. Definuje také své vlastní, nové třídy a API, ve snaze sjednotit vývoj pro mobilní zařízení a počítače.

Dobrým tahem je zachované a posílené provázání s XAML, které je zde využíváno pro vytvoření uživatelských rozhraní. XAML bylo velmi široce využíváno i v dřívějších verzích. Taktéž se vyskytuje ve Windows Presentation Foundation a Microsoft Silverlight, kde je, stejně jako ve WinRT, určeno k tvorbě UI prvků a jejich komunikaci s kódem na pozadí. Avšak ve Windows Phone 8 je Windows Phone XAML Framework založen na stejném frameworku, jako v Silverlight frameworku nebo jako ve Windows Phone 7. Windows Phone 8.1 a Windows 8/8.1 přináší Windows Runtime XAML Framework, což ve zkratce znamená lepší funkcionalitu a nové možnosti.

### 3.2 MVVM

Model View ViewModel je vzor architektury navržený společností Microsoft původně hlavně pro WPF aplikace. Dnes tento model vyhovuje také aplikacím vyvíjeným pro Windows Store a Windows Phone.

Tento návrhový vzor má tři vrstvy. Nejprve je zde Model. Jedná se o třídu nebo soubor tříd popisující data, se kterými aplikace pracuje. Dále je zde View, vrstva reprezentující uživatelská rozhraní. A nakonec ViewModel, který využívá jedné třídy, jež bude udržovat stav aplikace. Pomocí rozhraní INotifyPropertyChanged je možné pružně reagovat na změnu dat ve třídě ViewModel a okamžitě ji projevit také v uživatelském rozhraní.

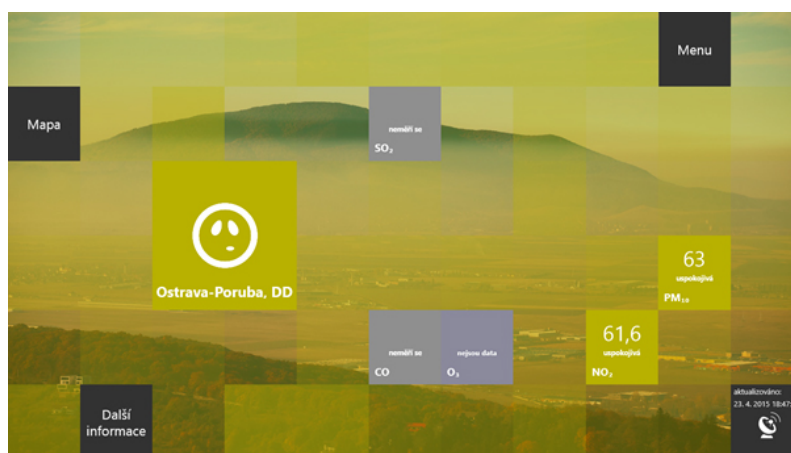


Tato propagace změn je realizována za pomoci vlastnosti data binding, která se vyskytuje v technologiích založených na jazyce XAML.

Vývoj aplikace Čistota ovzduší je založen na architektuře MVVM. Univerzální aplikace mají pro tento přístup uzpůsobenou strukturu.

### 3.3 Uživatelské rozhraní

Hlavní ideou pro vytvoření nového rozhraní bylo zachování stěžejních funkcí mobilní aplikace, ale zároveň poskytnutí uživateli rozšířené intuitivnosti a navázání na přirozené funkce prostředí Windows 8.1.

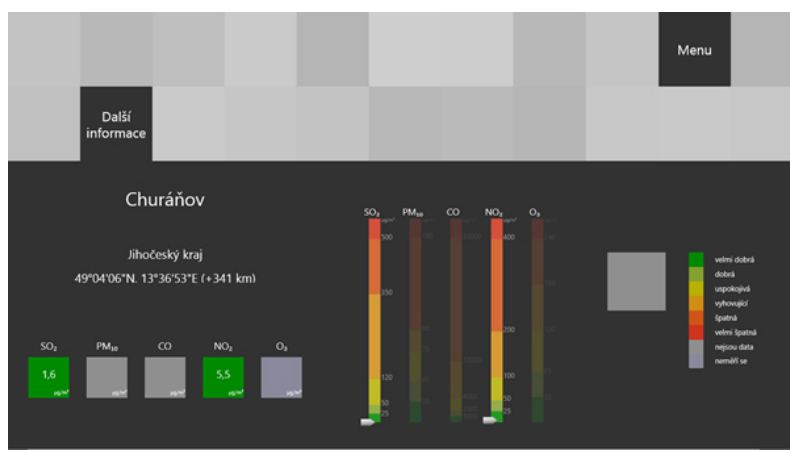


Obrázek 1: Úvodní stránka aplikace

Toto nové uživatelské rozhraní mělo splňovat zásady Modern UI, a to jednoduchost a zaměření se na obsah. Středobodem aplikace je hlavní dlaždice zobrazující aktuální stav stanice. Jedná se buď o nejbližší stanici, nebo o stanici zvolenou uživatelem. Tato možnost závisí na nastavení aplikace. Při zběžném pohledu uživatel okamžitě zjistí pouze nezbytné množství aktuálních informací, tedy celkový stav ovzduší a stavy jednotlivých měřených složek. Těmito složkami jsou oxid siřičitý, oxid dusičitý, oxid uhelnatý, přízemní ozon a polétavý prach. Také je zde umístěna informační dlaždice pro zobrazení času poslední aktualizace a stavu lokalizace. Viz obrázek č. 1.

Další možnosti interakce s aplikací jsou směřovány do tří panelů, které jsou vysunuty po kliknutí na příslušné tlačítko. Panel ve spodní části obrazovky poskytuje rychlý přehled o aktuálně zvolené či vyhledané stanici. Viz obrázek č. 2.

Ve druhém panelu, na levé straně obrazovky, se nachází mapa České republiky. Při vysunutí panelu se vykreslí slepá mapa Česka se stanicemi ČHMÚ (obr. č. 3). Uživatel



Obrázek 2: Informační panel

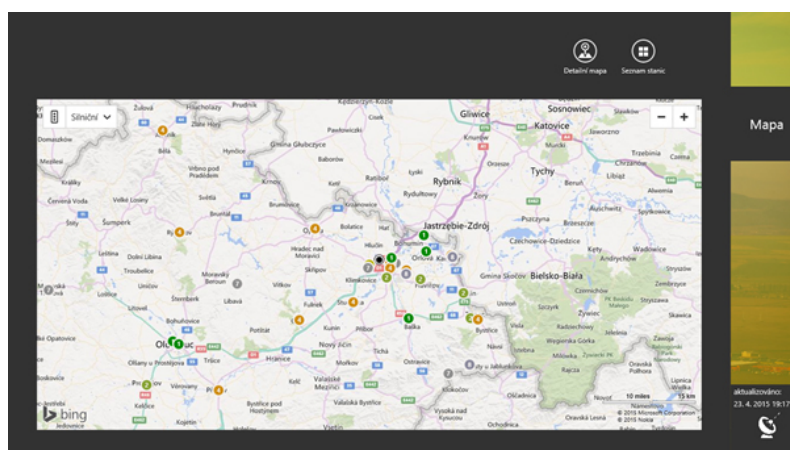
má možnost tuto slepou mapu kliknutím na tlačítko *Detailní mapa* zaměnit za pohled na Bing mapu. Může zde vidět taktéž rozmístěny ukazatele na stanice s příslušným zbarvením dle stavu a navíc je zde černobílý ukazatel uživatelské pozice, viz obrázek č. 4.



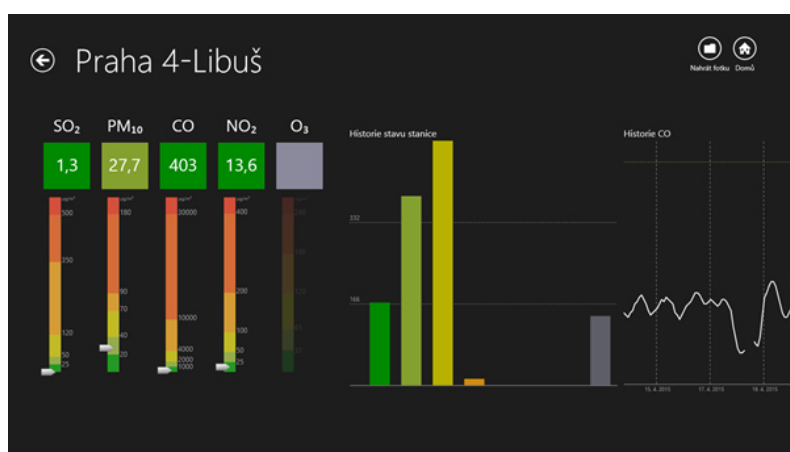
Obrázek 3: Panel mapy - slepá mapa

Poslední panel je umístěn v horní části obrazovky a tvoří menu aplikace. Skrze tento panel se uživatel může dostat k pokročilejším funkcím aplikace nebo do jejího nastavení.

Pokud si uživatel přeje prohlédnout detailnější údaje, statistiky, zobrazit hodnocení stanice ostatních uživatelů nebo přidat hodnocení vlastní, může být skrze menu nebo hlavní dlaždici přeměrován na stránku stanice, viz obr. č. 5. Stránky jednotlivých stanic, seznam stanic a nahrávání hodnocení jsou realizovány na samostatných stránkách.



Obrázek 4: Panel mapy - detailní mapa



Obrázek 5: Stránka stanice

Uživatelské rozhraní je koncipováno tak, aby zpřístupnilo a intuitivně poskytlo uživateli všechny důležité informace na úvodní stránce aplikace. V případě, že si uživatel přeje pokročilé informace nebo funkce, může je snadno nalézt na několika vedlejších stránkách.

Rozhraní je řešeno kombinací XAML a kódu na pozadí. Hlavní stránku aplikace jsem zpracoval převážně v code-behind. V době tvorby rozhraní jsem to považoval za nejlepší z důvodu využití mnoha možností, které pro uživatelské rozhraní Windows 8.1 nabízí a nad kterými lze mít v jazyce C# větší kontrolu, než v jazyce XAML. Zvláště pak, co se týče orientace obrazovky, animací nebo řešení samotných dlaždic, které jsou tvořeny čistě v code-behind.

Na dalších stránkách aplikace, které neobsahovaly tak složité skupiny elementů uživatelského rozhraní, jsem využil jazyka XAML v maximální míře a code-behind poté zahrnoval nezbytné obslužné metody. Ve výpisu kódu č. 1 je možné vidět element tvořený pouze XAML kódem. Tento element nemá žádný code-behind, využívá pouze dvou konventorů hodnot.

---

```
<StackPanel x:Name="statusSOStackPanel" Margin="0" Grid.Row="1">
  <Viewbox Width="80" Height="80" Margin="10">
    <Grid x:Name="statusSOClr"
      Background="{Binding_So2.State,Converter={StaticResource_
        ColorQualityConverter}}"
      Height="80"
      Width="80">
      <TextBlock x:Name="statusSOTxt"
        TextWrapping="Wrap"
        Text="{Binding_So2.Value,Converter={StaticResource_
          NegativeValueConverter}}"
        FontSize="30"
        Margin="0"
        VerticalAlignment="Center"
        HorizontalAlignment="Center"/>
    </Grid>
  </Viewbox>
</StackPanel>
```

---

Výpis 1: Element pro zobrazení hodnoty jednoho z prvků

Nastavení, informace o aplikaci a podmínky ochrany osobních údajů aplikace jsem zpracoval doporučeným způsobem. A to v postranním panelu operačního systému, kde jsou umístěny pod položkou *Nastavení*. Tyto elementy dědí ze šablony Flyout, která je z důvodu maximálního sjednocení uživatelského rozhraní Windows Store aplikací určena přesně k tomuto účelu.

### 3.4 Využití navigace

Jedním z hlavních aspektů aplikace je využití GPS navigace. Ve Windows Runtime je přístup k navigaci výrazně zjednodušen využitím třídy Geolocator. Tato třída využívá k získání dat několik kombinovaných zdrojů. V první řadě je to lokátor zařízení, potažmo driver GPS lokátoru zařízení. Dále je to Windows Location Provider, který využívá data získaná ze softwarových zdrojů, jako je IP adresa nebo Wifi triangulace. A dále jsou to zdroje, které se mohou vyskytovat u využitých platform. Windows Location Provider prohlédne všechny dostupné zdroje, zvolí nejspolehlivější z nich a přijme data, která pomocí vnitřních procesů převede na typově specifická data Geolocation, s nimiž se dá objektivně pracovat.

---

```

void locator_PositionChanged(Geolocator sender, PositionChangedEventArgs args)
{
    Geoposition position = null;
    var task = Task.Run(async () => {
        position = await sender.GetGeopositionAsync(
            new TimeSpan(0, 0, 0, 200),
            new TimeSpan(0, 0, 3));
    });
    task.Wait();

    MyGeocoordinate coord = new MyGeocoordinate();
    coord.Latitude = position.Coordinate.Point.Position.Latitude;
    coord.Longitude = position.Coordinate.Point.Position.Longitude;
    if (coord != null)
    {
        MyPosition = coord;
    }

    //rest of the code
}

```

---

#### Výpis 2: Část metody pro obsluhu události změny geografické polohy

Ve Windows Phone 8 je s lokalizací pracováno za pomoci třídy `GeoCoordinateWatcher`, jejíž instancí je hlídač pozice, který vrací objekt třídy `GeoCoordinate` při každé změně pozice zařízení. Koordináty lze snadno získat pomocí vlastností třídy `Geocoordinate`, `Latitude` a `Longitude`.

Při migraci kódu lokalizace jsem využil již zmiňované třídy `Geolocator`. Při změně pozice jsem získal instanci třídy `Geoposition`. Jelikož koordináty jsou v této instanci uloženy ve vlastnosti `Coordinate`, v ní ve vlastnosti `Point`, v ní ve vlastnosti `Position` a až zde ve vlastnostech `Latitude` a `Longitude`, vytvořil jsem vlastní třídu, která obsahovala pouze nutné vlastnosti. S instancí této vytvořené třídy jsem potom mohl využít původních algoritmů pouze s malou obměnou. Příklad lze vidět ve výpisu kódu č. 2.

Aplikace využívá navigace pro hledání pozice zařízení uživatele. Díky získané pozici lze vypočítat vzdálenost vůči všem stanicím a zobrazit nejbližší stanici na úvodní obrazovce aplikace. Díky možnostem poskytnutým třídou `Geolocator` je zobrazen i stav geolokace v podobě piktogramu na informační dlaždici.

### 3.5 Úlohy na pozadí a živé dlaždice

Aplikace Čistota ovzduší obsahuje také funkci `Live Tile`. Jedná se o princip živé dlaždice, která nezávisle na stavu samotné aplikace zobrazuje určitá data. Nejlepším příkladem může být zobrazení počtu přijatých e-mailů i při vypnutém poštovním klientu nebo zobrazení fotek z fotoalba na dlaždici.



Nejprve jsem pracoval na formátu dat zobrazovaných dat. Na Windows Phone 8 bylo možné vytvořit vlastní živou dlaždici jako UserControl. Tedy vytvoření vlastního prvku uživatelského rozhraní, který je nadále možné využívat jako běžný element. Tato možnost byla zrušena a aktuálně není pro Windows 8.1 a pro Windows Phone 8.1 přístupná.

Vlastní živé dlaždice nahradila třída `TileTemplateType`. Třída poskytuje množství šablon, které můžeme pro dlaždice využít. Tyto šablony mají vnitřní strukturu tvořenou XML, a proto se snadno dají naplnit daty za pomoci tříd `XmlDocument`, `TileNotification` a `TileUpdateManager`. Viz výpis kódu č. 3.

V Modern UI existují čtyři velikosti dlaždic. První je nejmenší, čtvercová dlaždice, kterou není možné naplnit jakýmkoliv obsahem. Tato malá dlaždice je automaticky osazena logem aplikace vloženým do manifestu aplikace. Druhá dlaždice je čtverec o velikost strany 150 pixelů. Třetí dlaždice je obdélník o výšce 150 pixelů a šířce 310 pixelů. Čtvrtá dlaždice je největší možný rozměr, čtverec o délce strany 310 pixelů. Poslední tři vyjmenované jsou dostatečně velké pro zobrazení obsahu. Existují pro ně různé typy šablon k umístění textu a obrázků. Při využití více rozměrů dlaždic je nutné povolit frontu, která bude udržovat obsah k naplnění těchto dlaždic. Výchozí nastavení obsluhy dlaždic totiž pracuje s lineárním přístupem, kdy je obsah umístěn pouze do největší z povolených dlaždic.

---

```

TileUpdateManager.CreateTileUpdaterForApplication().EnableNotificationQueueForSquare150x150
(true);

XmlDocument contentSmall = TileUpdateManager.GetTemplateContent(TileTemplateType.
    TileSquare150x150PeekImageAndText04);
contentSmall.GetElementsByTagName("text")[0].InnerText = stationName;
contentSmall.GetElementsByTagName("image")[0].Attributes[1].InnerText = stationImage;

TileNotification notifSmall = new TileNotification (contentSmall);
TileUpdateManager.CreateTileUpdaterForApplication().Update(notifSmall);

```

---

### Výpis 3: Naplnění a aktualizace dlaždice

Pro vytváření obsahu těchto dlaždic a jejich obsluze bylo možné využít pouze Windows Runtime Background Tasks a tedy třídu `BackgroundTask`, nahrazující třídu `ScheduledTaskAgent` používanou ve vývoji aplikací na Windows Phone 8. Tyto procesy na pozadí mají ve WinRT jasná pravidla, která musí při svém provozu splňovat. Musí být registrovány v manifestu aplikace a poté za běhu samotné aplikace pomocí kódu, čímž jsou aktivovány a spouštěny systémovými spínači nebo časovými intervaly.

V úloze na pozadí jsem se snažil využít již zmiňované lokalizace k určení polohy a zjištění nejbližší stanice nebo stanice označené uživatelem. Poté získat údaje o této stanici dotazem na server, data zpracovat a vložit do živé dlaždice. Ukázala se býti překážkou omezení, kterými jsou úlohy na pozadí vázány. WinRT rozlišuje dva základní typy

těchto úloh podle toho, zda mohou využívat zamykací obrazovku operačního systému nebo ne. Počet aplikací, které mohou obdržet toto oprávnění v OS Windows 8.1, je omezen na sedm. Hlavním rozdílem mezi těmito dvěma typy je vymezený procesorový čas a interval, v jakém jej obdrží. První typ, privilegované úlohy obdrží každých 15 minut dvě sekundy procesorového času, druhý typ obdrží jednu sekundu každé dvě hodiny [3].

Kód, který bylo nutné spustit v úloze na pozadí, nemohl být vykonán v tak krátkém čase. Jedinou zbývajícím možností bylo využití další nové funkce, která přišla s aktualizací 8.1, tzv. Push Notifications. Jedná se o data vnučená aplikaci prostřednictvím serveru. Bylo ovšem rozhodnuto o nevyužití této možnosti a živá dlaždice nebyla zahrnuta do finální verze.

### 3.6 Komunikace

Aplikace stahuje v pravidelných intervalech ze serveru řetězec stanic s jejich vlastnostmi a ukládá jej jak do View Modelu, tak do souboru. Po stažení je řetězec rozparsován a uložen již jako seznam objektů opět do View Modelu. Dále se s nimi pracuje dle potřeby.

Stahování dat je zde realizováno pomocí třídy HttpClient, která nahrazuje třídu WebClient a je až na malé změny její kopií. Pomocí algoritmu přejatého z výchozí aplikace je vytvořen řetězec klíče, který je odeslán na server a zpracován jako dotaz na databázi. Takto jsou získávána jak samotná data stanic, tak fotografie, které jsou stahovány až při navigaci na stránku stanice.

Další komunikace se serverem probíhá při odeslání fotografie a komentáře uživatelem. Při odeslání je pomocí algoritmu složen proud odpovědi, který obsahuje fotografii, hodnocení stavu, identifikátor uživatele, volitelně komentář uživatele a další data.

### 3.7 Využití vlastností architektury MVVM a jazyka XAML

V aplikaci na mnoha místech využívám možností, které jazyk XAML ve spojení se zvolenou architekturou nabízí. Jedná se především o data binding, tedy svázání dat s určitým prvkem uživatelského rozhraní. Tato vlastnost umožňuje bez komplikací a dodatečných obslužných metod aktualizovat okamžitě při změně dat veškeré jejich příjemce.

Aby aktualizace daného elementu uživatelského rozhraní v UI vlákne proběhla, je nutné, aby vlastnosti datového kontextu nebo zdroje dat, v tomto případě třídy ViewModel, využívaly metod rozhraní INotifyPropertyChanged, které propagují změnu všem zúčastněným. Je také nutné, aby tato změna dat proběhla v UI vlákne. Znamená to, že každá taková operace, která probíhá v jiném vláknu, než v UI vláknu, musí být přiřazena UI vláknu prostřednictvím metody Dispatcher.RunAsync, viz výpis kódu č. 4.

---

```
CoreApplication.MainView.CoreWindow.Dispatcher.RunAsync(CoreDispatcherPriority.Normal, () =>
{
    _gpsService.SetPosition();
});
```

---

Výpis 4: Přiřazení části kódu UI vláknu

Za předpokladu splnění všech podmínek jsou změny dat promítnuty okamžitě, což lze v aplikaci pozorovat například v nastavení. Pokud uživatel otevře nastavení, je mu vždy zobrazena aktuálně zvolená stanice, bez ohledu na nastavení automatického vyhledávání stanic. Při výběru stanice ze seznamu stanic je v třídě View Model změněna vlastnost CurrentStation. Tato operace se za pomoci data bindingu promítne do celé aplikace a je okamžitě viditelná na úvodní obrazovce.

### 3.8 Práce s vlákny a asynchronními operacemi

Na rozdíl od Windows Phone 8 se zde nedá mluvit příliš o práci s vlákny, jenž byla potlačena. Nahradila je třída Task a princip async await jazyka C# 5.0, které je zastupují. Místo konvenční třídy Thread zde nalézáme třídu Task, jež se používá podobně. Hlavní rozdíl je v manipulaci s asynchronními metodami různých systémových tříd a API, stejně jako s vlastními asynchronními metodami vytvořenými za pomoci třídy Task, která probíhá právě za pomoci klíčových slov async a await [4].

Mnoho metod použitých v aplikaci běží asynchronně. Jedná se jak o vlastní metody, tak o metody tříd, jako je HttpClient. Také s některými prvky uživatelského rozhraní je nutno pracovat asynchronně. Je to například upozornění, MessageDialog.

Metoda, v níž je použita jakákoliv asynchronní metoda spouštěná s klíčovým slovem await, musí být v deklaraci označena slovem async a znamená to, že se jedná o kompozici asynchronních operací. Metoda označena jako async musí dle dokumentace vracet datový typ Task<T> nebo Void. V průběhu vývoje aplikace se druhá možnost ukázala být problematickou a nepoužitelnou. V případě, že metoda nevrací žádnou hodnotu, její návratový typ musí být deklarován jako Task<void>, případně jako Task, viz výpis kódu č. 5. V opačném případě může dojít k nepředvídatelnému chování programu.

---

```
public async Task LoadData()
{
    //method implementation
}
```

---

Výpis 5: Korektní hlavička asynchronní metody

Problém nastal v kombinaci asynchronních operací a data binding. Asynchronní metody běží typicky v pracovním vlákne. Pokud se ale pokusí ovlivnit data třídy ViewModel, která jsou pomocí metody NotifyPropertyChanged okamžitě propagována do uží-

vatelského rozhraní, dojde ke křížení vláken. Je tedy nutné změnu inkriminovaných dat uvnitř asynchronních metod vztahovat k UI vláknu.

Asynchronní operace jsou využity napříč celou aplikací. Od prvotního stažení dat a práci se soubory, přes výpočty, aktualizace stavů, odesílání uživatelského hodnocení, až po vyvolávání oken upozornění.

### 3.9 Dokončení aplikace

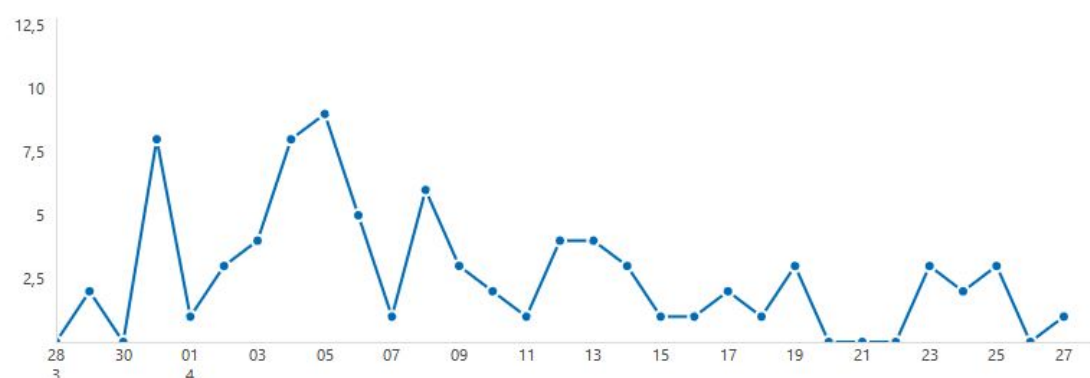
Po dokončení implementační části aplikace bylo potřeba vyplnit manifest aplikace. Vzhledem k multijazyčnosti aplikace jsem musel název zadat přímo v kódu manifestu v podobě odkazu na jazykové soubory. Pro Windows Universal Apps je povinné vložit grafické materiály pro prezentaci aplikace na Windows Store, zobrazení dlaždic, úvodní obrazovku aplikací atd. Dále jsem musel definovat možnosti aplikace, nebo-li práva na využívání zdrojů. V tomto případě se jednalo o webkameru, připojení k internetu a polohu zařízení. Po zadání jména společnosti jako vydavatele aplikace jsem předal balíček aplikace firmě.

Aplikace byla nasazována na Windows Store. V této fázi se vyskytly potíže s certifikací. I přes funkčnost aplikace ve vývojovém prostředí byla aktualizace opakovaně odmítána. Ladění bylo složité i proto, že jsem z bezpečnostních důvodů neměl přístup k vývojářskému účtu firmy, určenému k nasazování aplikací a chyby, nahlášené certifikační prověrkou, jsem ladil zprostředkovaně. Využíval jsem tedy program Windows App Cert Kit, jenž je určen k otestování aplikace před samotným pokusem o nasazení. Není ovšem úplně spolehlivý.

Aplikaci se podařilo umístit na Windows Store, kde je k dispozici ke stažení. Na obrázku č. 6 je možné vidět statistiku stahování aplikace uživateli za období 28. 3. 2015 – 27. 4. 2015.

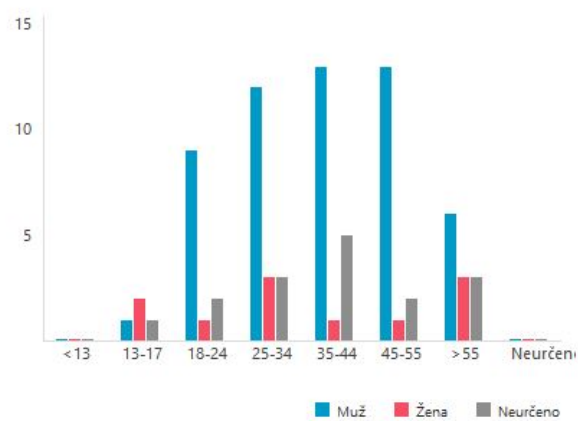
### Počet stažení

Počet stažení aplikace

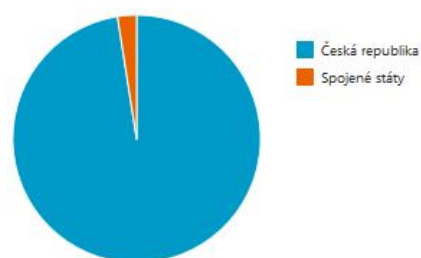


### Počet stažení podle věkové skupiny

Počet stažení aplikace



### Počet stažení podle trhu



Obrázek 6: Statistika stahovanosti



## 4 Poznámky k platformě a zhodnocení znalostí

I když jsem již před samotnou praxí s Windows Runtime pracoval, nikdy jsem nevyužil takový rozsah jeho možností. V průběhu praxe jsem tedy došel k mnoha poznatkům o tomto frameworku. Univerzální aplikace šetří práci a při použití architektury MVVM tvoří jasně danou cestu, kterou by se měl vývoj aplikace ubírat. Tato architektura je jasně navržena pro plné využití možností jazyka XAML. Což jsou například přehledná struktura a dobře zpracovaný nástroj k tvorbě uživatelského prostředí ve Visual Studio. Naproti tomu Visual Studio 2013 nenabízí při ladění kódu v jazyce XAML takový komfort, jako při ladění code-behind.

Bohužel však to, co tato platforma nabízí zjednodušením práce s UI a různými specifickými API, to ji schází při práci s asynchronními operacemi. Ty mohou být při větší složitosti aplikace, bez naprosto jasné představy a pochopení syntaxe, velmi matoucí a zapříčinit chybovost aplikace. Tento systém práce s asynchronními operacemi nedává programátorovi takovou kontrolu, jakou poskytovala konvenční vlákna v minulých verzích.

Windows Universal Apps jsou dobrým způsobem pro vývoj menších a středních aplikací, vhodným i pro méně zkušené programátory, kteří mohou využít předpřipravených šablon pro multiplatformní vývoj. Ve spojení s návrhovým vzorem MVVM udávají jasný směr, kterým by měl vývoj aplikace jít. Mnou vytvořená aplikace byla určena pouze pro desktop a bylo možné využít některé ze šablon Windows Store aplikací, použití Universal App ovšem poskytuje základ pro případnou aktualizaci současné mobilní verze, a také při případné migraci kódu na Windows 10, jehož příchod je očekáván v druhé polovině roku 2015. Co se týče Windows Runtime, osobně jej hodnotím jako vynikající a moderní nástroj pro vývoj méně složitých aplikací. Není naopak příliš vhodný k programům využívajícím složitých algoritmů a pokročilé práce s asynchronními operacemi za předpokladu, že vývojář není opravdu zkušeným programátorem zběhlým ve WinRT.

### 4.1 Uplatněné znalosti

V průběhu odborné praxe jsem využil znalosti a zkušenosti získané převážně v průběhu těchto předmětů:

Algoritmy II – Tento předmět mi osvětlil zákonitosti a problematiku objektově orientovaného programování.

Programovací jazyky II – V tomto předmětu jsem se naučil základům jazyka C#, jehož jsem využil při vývoji aplikace Čistota ovzduší.

Vývoj informačních systémů – Předmět mě uvedl do problematiky návrhových vzorů, architektur a dovedl mě k pochopení mechanismů důležitých pro běh životního cyklu

programů. Zkušeností získaných zde jsem využil i při vývoji aplikace.

Architektura technologie .NET – Díky tomuto předmětu jsem získal znalosti o frameworku .NET a jeho využití. Pomohlo mi to pro získání přehledu o rozdílech v programování pro starší verze Windows Phone a WPF, ve srovnání s WinRT, používaným v nejnovějších aplikacích.

Databázové a informační systémy – V tomto předmětu jsem se naučil pracovat se sítí a pochopil jsem komunikaci mezi rozdílnými platformami. Tyto znalosti mi pomohly nejen při výkonu odborné praxe.

Tvorba aplikací pro mobilní zařízení I – Předmět mě, víc než co jiného, přivedl k tématice aplikací pro mobilní zařízení a vděčím mu za motivaci věnovat se jí i ve volném čase.

Informační systémy pro elektronické publikování – Díky tomuto předmětu jsem se dostal k vývoji aplikací na Windows Phone, čemuž jsem se ve volném čase učil a což mi pomohlo pochopit a využít pokročilejší možnosti vývoje aplikace v rámci odborné praxe.

## 4.2 Scházející a nabyté znalosti

Na začátku odborné praxe jsem neměl příliš rozsáhlé znalosti Windows Runtime, ve kterém bylo nutné aplikaci Čistota ovzduší vyvíjet. Měl jsem jisté zkušenosti s vývojem na Windows Phone 7.5, který ale probíhal ve starších verzích jazyka C# a technologie .NET. Práce s Universal App pro mě tedy byla do jisté míry neznámou. Také mi chyběly hlubší programátorské znalosti a porozumění problematice. I když jsem měl základy, vývoj této aplikace pro mne byl cestou plnou vzdělávání se, řešení problémů a učení se z nich.

Před samotným vývojem jsem dostal k dispozici zdrojový kód aplikace již používané pro mobilní telefony. Do té doby jsem neměl co dočinění s kódem tak rozsáhlým a bylo to pro mne dost matoucí. Zvláště vzhledem k absenci dokumentace.

V té době jsem měl již omezené znalosti jazyka XAML, kterých jsem využil a ač jsem jej ze začátku nepovažoval za efektivní, po absolvování této praxe jsem změnil názor a jazyk zařadil mezi své favority. Sám jsem se přesvědčil o jeho efektivitě a jednoduchosti, jakmile do něj člověk jednou pronikne.

Podobným způsobem jsem si oblíbil WinRT využívající nejnovější standardy, které práci v mnohém zjednodušují. I když v rámci zjednodušení jsou okleštěny původní mechanismy, což v konečném důsledku může přinést ještě více práce a problémů s laděním aplikace.

Při zpětném pohledu na vývoj aplikace, který probíhal v průběhu 5. a 6. semestru, vidím, že jsem získal ohromné množství zkušeností. Dnes bych aplikaci zpracoval v mno-

hém jinak. Efektivněji, přehledněji, a bezesporu mnohem rychleji. Po technické stránce jsem nabyl mnoho vědomostí, týkajících se WinRT, C#, XAML, práce s Visual Studio 2013 a Blend. Po osobnostní stránce jsem získal lepší pracovní morálku, jelikož jsem se musel naučit i v domácím prostředí být plně soustředěný na úkol.

## 5 Závěr

Odborná praxe mi dala mnoho zkušeností. Mnohem více, než by mi mohla dát typická bakalářská práce. Pracoval jsem pod odborným dohledem na úkolech a cílech, jež mi byly zadány. Pravidelně jsem konzultoval své výsledky a pokroky s konzultantem, který mi udával směr práce.

Naučil jsem se řešit problémy a pracovat samostatněji. Získal jsem lepší pracovní morálku a zvykl jsem si mít zodpovědnost za svou práci. Aplikace Čistota ovzduší, kterou jsem během této praxe vyvíjel, je aktuálně k dispozici ke stažení zdarma na Windows Store pro systémy Windows 8 a Windows 8.1.

## 6 Reference

- [1] *GARVIS Solutions* [online]. © 2013 [2015-04-20]. Dostupné z: <<http://garvis.cz>>
- [2] *API changes for Windows 8.1(XAML)* [online]. © 2015 [2015-04-20]. Dostupné z: <<https://msdn.microsoft.com/cs-cz/library/dn263110.aspx>>
- [3] *Introduction to Background Tasks* [online]. Ver. 1.2, 5.10.2012 [2015-04-20]. Dostupné z: <<http://www.microsoft.com/en-us/download/details.aspx?id=27411>>
- [4] *Quickstart: Calling asynchronous APIs in C# or Visual Basic* [online]. © 2015 [2015-04-20]. Dostupné z: <<https://msdn.microsoft.com/en-us/library/windows/apps/hh452713.aspx>>